Proyecto Júpiter - Informe Ejecutivo

DETECCIÓN DE FAKE NEWS

Pontiapp

Pontia - Octubre 2024

ÍNDICE

1. **Equipo del proyecto y objetivos:**

En este proyecto hemos trabajado como unidad y equipo. Los objetivos fijados por la empresa, son encontrar una sistema adecuado capaz de identificar el fraude y transformar los datos digitalmente, así como el cálculo de KPIs útiles para el negocio.

- Lucía Cámara Chamorro
- Amparo Díaz Román
- María Belén Libonati Cattai
- Carolina Méndez

JSON original:

- t_id: identificador único de la noticia.
- tiempo: unidad de tiempo (número entero).
- título: título con el que se identifica la noticia.
- texto: texto íntegro de la noticia.
- fake: naturaleza de la noticia clasificada como: fake new (FAKE) o noticia veraz (REAL).
- autor: autor que redacte la noticia.
- fuente: periódico o página web que publique el artículo bajo su firma.
- tipo: tipo de noticia (columna, carta al editor, artículo de opinión, …).
- visitas: número de visualizaciones que ha recibido la noticia.
- compartir: número de veces que se ha compartido el enlace de la noticia en redes sociales.
- compartir_tiempo: al igual que en el campo de tiempo, unidad de tiempo (número entero) que representa el momento en el que se comparte el artículo periodístico.
- duración: tiempo de lectura del usuario.
- favorito: número de veces que un usuario ha marcado el artículo como favorito.
- país: país donde se publica la noticia.
- idioma: idioma en el que se redacta la noticia.

2. **Limpieza de datos:**

Comprobación de nulos:

- Categóricas: se han cambiado por una cadena de caracteres. Ejemplo: nulos en autor a “sin autor”.
- Numéricas: sustituidos por la media según *tipo* de noticia.

En el resto de columnas no se han realizado cambios, ya que contienen información útil.

Duplicados en columnas: Se comprueba primero si tienen sentido o no, en las columnas *texto* y *título* no tienen sentido. Se crea un nuevo Data Frame sin duplicados para usarlo en los modelos y evitar confusiones. Se dejan los duplicados en el df original para la base de datos.

Registros duplicados no encontramos.

Las fechas de publicación posteriores a fecha de compartido, damos por hecho que son datos que se han introducido de forma errónea, así que se intercambian.

En las columnas *Textos* y *títulos*, se limpia de espacios en blanco al inicio y al final. Comprobamos que no hay caracteres no imprimibles. Eliminamos todo tipo de enlaces, imágenes, url, etc.

En las columnas numéricas se eliminan valores con caracteres que impiden el cambio de tipo de la columna.

Comprobamos duplicados en *id_noticia* y no hay.

Encontramos valores negativos en *duración* y se pasan a valor absoluto, ya que entendemos que es un error. Al igual que la *duración*, estaba en segundos y cambiamos a minutos.

Dentro de la columna de *tipo*, se pueden unificar estas dos categorías "Entrevista en profundidad" y “Entrevista a fondo”.

Errores e incidencias fijadas por la empresa :

- Una columna no puede tener más de 2.500 caracteres.
- El tiempo de lectura no puede ser mayor a 20 minutos.
- No se publican más de 50 noticias al día.
- Una misma fuente no publica noticias en más de dos idiomas distintos.

Durante la limpieza de datos encontramos muchas noticias que tenían el *título* igual al *texto*, y tras comprobar ambas noticias, detectamos que eran falsas.
En las columnas numéricas encontramos errores que pasamos a Nan, eliminamos los valores negativos. Con los nulos procedemos al cambio por su mediana según el tipo y creamos una nueva columna “medianas”, para posteriormente proceder al cambio a numérico.
Durante la limpieza de los datos, nos dimos cuenta que el mayor porcentaje de Fake News son las noticias de última hora, ya que no da tiempo de contrastar dicha veracidad cuando ya ha sido divulgada.

3. Modelo relacional:

Se desarrolla un modelo de datos utilizando los documentos CSV obtenidos luego de la limpieza de datos, para este modelo los datos se dividen en 6 tablas según lo detallado a continuación, de forma que sencillamente es posible realizar consultas para obtener KPIs y responder a las más frecuentes preguntas de negocio.
Se decide extraer la columna “contenido” de la tabla de hechos, aunque su primary key es el mismo que el de la tabla “noticias” (id_noticia), con el fin de eficientizar el almacenamiento y las consultas, debido a la cantidad de almacenamiento que ocupa y a que no es relevante en gran parte de KPIs/preguntas.

Tablas:

TABLA DE HECHOS:

noticias - Primary Key: id_noticia | Foreign Key: id_tipo / id_pais_idioma / id_fuente / id_autor

TABLAS DE DIMENSIONES:

autor - Primary Key: id_autor

contenido - Primary Key: id_noticia
fuente - Primary Key: id_fuente
pais_idioma - Primary Key: id_pais_idioma
tipo - Primary Key: id_tipo

Tipos de datos:

- Numéricos →

  SMALLINT/MEDIUMINT para optimizar el almacenamiento al tratarse de números enteros entre pequeños y medianos de largo máximo predecible.

  BOOLEAN al cargar "veracidad" como 0/1, aunque luego es convertido a texto para facilitar la comprensión de las consultas.

- Textos →

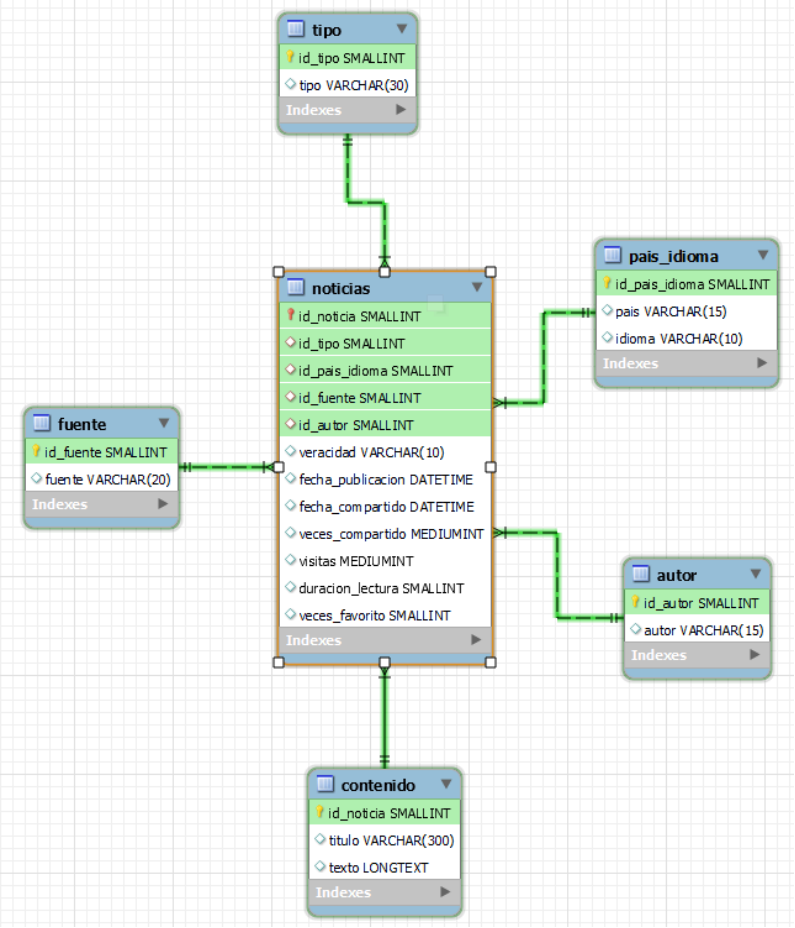
  VARCHAR usado para almacenar cadenas de texto cortas y medianas de longitud máxima predecible,es más eficiente que TEXT en estos casos y específicamente sabiendo que se realizarán numerosas consultas referidas a estas columnas.

  LONGTEXT en el caso de los textos de las noticias ya que TEXT no es opción suficiente, y esta permite hasta 4,294,967,295 caracteres.

- Fechas→

  DATETIME para almacenar fechas y horas exactas en las que las noticias se comparten y publican, sin cambios según zona horaria.

Esquema de la Base de datos relacional:



4. Metodología de Machine Learning:

En este caso se elige un aprendizaje supervisado pues tenemos un histórico de datos etiquetados, noticias que ya han sido contrastadas y catalogadas, que se usa para entrenar modelos de clasificación binaria para que clasifiquen nuevas noticias entre dos categorías, real y falsa.

Preprocesamiento de los datos: Se preparan los datos para tener un formato adecuado para los algoritmos y para ello se sigue un pipeline o serie de procesos.

- Codificación: las variables numéricas no necesitan de codificación ya que consiste en obtener valores numéricos que son los adecuados para introducir en los modelos, las variables categóricas como autor, tipo, país y fuente se transforman a numéricas usando one hot encoding. La variable objetivo o etiquetas de nuestras noticias se codifican como 0 para falso y 1 para verdadero.
- Obtención de los grupos de entrenamiento y prueba: se separa el histórico de datos en dos grupos, un 85% pasa a ser el grupo de train o entrenamiento y el 15% restante el grupo de test o pruebas. Se necesita que el grupo de entrenamiento sea grande para poder entrenar bien los modelos, cuantos más datos reciban los modelos, más casos distintos pueden aprender.
- Tokenización y embedding: Los textos del contenido de las noticias y el título de las noticias se codifican mediante tokenización para separar los textos en token o palabras (normalmente suelen ser palabras) y convertir estos tokens en un formato que un algoritmo pueda leer, numérico. Tras esto, se convierten en embeddings que divide esos tokens en varios componentes para un procesamiento del texto de mayor precisión.

Selección de algoritmo: Para este caso se prueban varios modelos simples y avanzados de machine learning además de modelos de deep learning para ver cuál devuelve una mayor precisión en la evaluación con el grupo de test.

- Modelos simples: Regresión logística, Knneighbors, Support Vector Machine, Árbol de decisión
- Modelos avanzados, ensemblers: Bagging con tocones, Bagging con árboles de decisión, RandomForest
- Modelos deep learning: Red neuronal recurrente, Transformer, Red convolucional, LSTM, GRU.

Entrenamiento y parametrización del algoritmo: Se usa el mismo conjunto de datos de train para entrenar todos los modelos además de añadir una semilla que permite que los modelos devuelvan el mismo resultado en cada ejecución. Se utilizan técnicas de Grid Search con validación cruzada para entrenar un mismo modelo varias veces con diferentes parámetros y así encontrar los óptimos.
Para las redes neuronales se prueba con diferentes arquitecturas variando parámetros y número de capas para aumentar o disminuir la complejidad del modelo y dar con una mayor precisión.

Evaluación del modelo: Se calcula la pérdida comparando las predicciones del modelo y las etiquetas o valores reales. Con ello se miden métricas propias de problemas de clasificación como la precisión (Accuracy o precisión global), la especificidad (True Negative Rate o TNR), la sensibilidad (Recall, True Positive Rate o TPR), la matriz de confusión (que nos da una visión de las predicciones del modelo enfrentadas a los casos reales de los datos) y el área bajo la curva o curva ROC. Éstas métricas se calculan para el grupo de entrenamiento (train) y para el de prueba (test) y se comparan para

comprobar si el modelo sobre aprende (aprende demasiado de los datos de entrenamiento lo que lleva a tener mayor precisión que en el grupo de prueba) o generaliza demasiado (no aprende lo suficiente y las precisiones de train y test son bajas)

5. Comparación de modelos:

Según los resultados se eligen modelos que tengan mayor precisión global teniendo en cuenta que el TNR y el Recall estén balanceados dando preferencia a tener un mayor TNR, esto se debe a que se busca mitigar lo que se llama error de tipo 1 que consiste en los falsos negativos que el modelo ha predicho, o dicho de otra forma, las noticias falsas que el modelo ha clasificado como verdaderas.

Modelo	Precisión	TNR	Recall
Regresión logística	68,3%	55,4%	82,2%
KNNeighbors	59,6%	53,3%	66,5%
Support Vector Machine	73,1%	73,3%	72,7%
Árbol de decisión	72,2%	67,4%	77,4%
Bagging de tocones	70,6%	53,3%	89,4%
RandomForest	77,9%	75,9%	80,1%
Bagging de árboles	78,5%	77,8%	79,2%
LSTM	57,9%	88,1%	25,2%
GRU	63,7%	72,7%	54,0%
Red neuronal convolucional	68,8%	63,1%	75,1%
Red neuronal recurrente solo textos	65,0%	74,0%	55,2%
Transformer Encoder	66,0%	59,5%	73,0%

6. Conclusiones y Líneas futuras:

Pontiapp requería un sistema adecuado de gestión y almacenamiento de datos y de identificación de noticias falsas, con el objetivo de poder proporcionar a sus clientes o usuarios un uso seguro, sano y confiable de su servicio. Con la realización de este proyecto se han logrado importantes avances en este sentido, destacando las siguientes aportaciones: a) Manejo adecuado de incidencias y errores, dando respuesta a los deseos comunicados por la empresa; b) identificación de parámetros de detección de noticias falsas; c) presentación de un modelo de machine learning y uno de deep learning con precisión aceptable para la categorización de noticias en reales o falsas; y d) propuesta de posibles aplicaciones de la IA generativa en línea con los objetivos de la empresa. En definitiva, el proyecto presentado responde a las necesidades de la empresa, ofreciendo no solo posibles soluciones para sus demandas actuales, sino también proponiendo posibles vías de actuación que mejoren el atractivo de la empresa para sus potenciales clientes y su competitividad en el mercado nacional e internacional. No obstante, si hubiéramos tenido más tiempo podríamos haber encontrado un modelo más preciso para la detección de noticias falsas, así como haber probado un modelo de natural language processing (NLP) preentrenado. También se podría haber realizado alguna prueba preliminar para el desarrollo de la herramienta propuesta de IA generativa dirigida al filtrado inteligente y resumen personalizado de noticias (TailorNews).